17.1 Basic configuration

The SPI is configured using the following registers:

1. Power: In the PCONP register (Table 46), set bit PCSPI.

   Remark: On reset, the SPI is enabled (PCSPI = 1).

2. Clock: In the PCLKSEL0 register (Table 40), set bit PCLK_SPI. In master mode, the clock must be an even number greater than or equal to 8 (see Section 17.7.4).
3. Pins: The SPI pins are configured using both PINSEL0 (Table 80) and PINSEL1 (Table 81), as well as the PINMODE (Section 8.4) register. PINSEL0[31:30] is used to configure the SPI CLK pin. PINSEL1[1:0], PINSEL1[3:2] and PINSEL1[5:4] are used to configure the pins SSEL, MISO and MOSI, respectively.
4. Interrupts: The SPI interrupt flag is enabled using the S0SPINT[0] bit (Section 17.7.7). The SPI interrupt flag must be enabled in the NVIC, see Table 50.

Remark: SSP0 is intended to be used as an alternative for the SPI interface. SPI is included as a legacy peripheral.

17.2 Features

- Compliant with Serial Peripheral Interface (SPI) specification.
- Synchronous, Serial, Full Duplex Communication.
- SPI master or slave.
- Maximum data bit rate of one eighth of the peripheral clock rate.
- 8 to 16 bits per transfer.

17.3 SPI overview

SPI is a full duplex serial interface. It can handle multiple masters and slaves being connected to a given bus. Only a single master and a single slave can communicate on the interface during a given data transfer. During a data transfer the master always sends 8 to 16 bits of data to the slave, and the slave always sends a byte of data to the master.

17.4 Pin description

Table 359. SPI pin description

Pin Name	Type	Pin Description
SCK	Input/ Output	**Serial Clock.** The SPI clock signal (SCK) is used to synchronize the transfer of data across the SPI interface. The SPI is always driven by the master and received by the slave. The clock is programmable to be active high or active low. The SPI is only active during a data transfer. Any other time, it is either in its inactive state, or tri-stated.
SSEL	Input	**Slave Select.** The SPI slave select signal (SSEL) is an active low signal that indicates which slave is currently selected to participate in a data transfer. Each slave has its own unique slave select signal input. The SSEL must be low before data transactions begin and normally stays low for the duration of the transaction. If the SSEL signal goes high any time during a data transfer, the transfer is considered to be aborted. In this event, the slave returns to idle, and any data that was received is thrown away. There are no other indications of this exception. This signal is not directly driven by the master. It could be driven by a simple general purpose I/O under software control.
MISO	Input/ Output	**Master In Slave Out.** The SPI Master In Slave Out signal (MISO) is a unidirectional signal used to transfer serial data from an SPI slave to an SPI master. When a device is a slave, serial data is output on this pin. When a device is a master, serial data is input on this pin. When a slave device is not selected, the slave drives the signal high-impedance.
MOSI	Input/ Output	**Master Out Slave In.** The SPI Master Out Slave In signal (MOSI) is a unidirectional signal used to transfer serial data from an SPI master to an SPI slave. When a device is a master, serial data is output on this pin. When a device is a slave, serial data is input on this pin.

17.5 SPI data transfers

Figure 75 is a timing diagram that illustrates the four different data transfer formats that are available with the SPI port. This timing diagram illustrates a single 8-bit data transfer. The first thing you should notice in this timing diagram is that it is divided into three horizontal parts. The first part describes the SCK and SSEL signals. The second part describes the MOSI and MISO signals when the Clock Phase control bit (CPHA) in the SPI Control Register is 0. The third part describes the MOSI and MISO signals when the CPHA variable is 1.

In the first part of the timing diagram, note two points. First, the SPI is illustrated with the Clock Polarity control bit (CPOL) in the SPI Control Register set to both 0 and 1. The second point to note is the activation and de-activation of the SSEL signal. When CPHA = 0, the SSEL signal will always go inactive between data transfers. This is not guaranteed when CPHA = 1 (the signal can remain active).

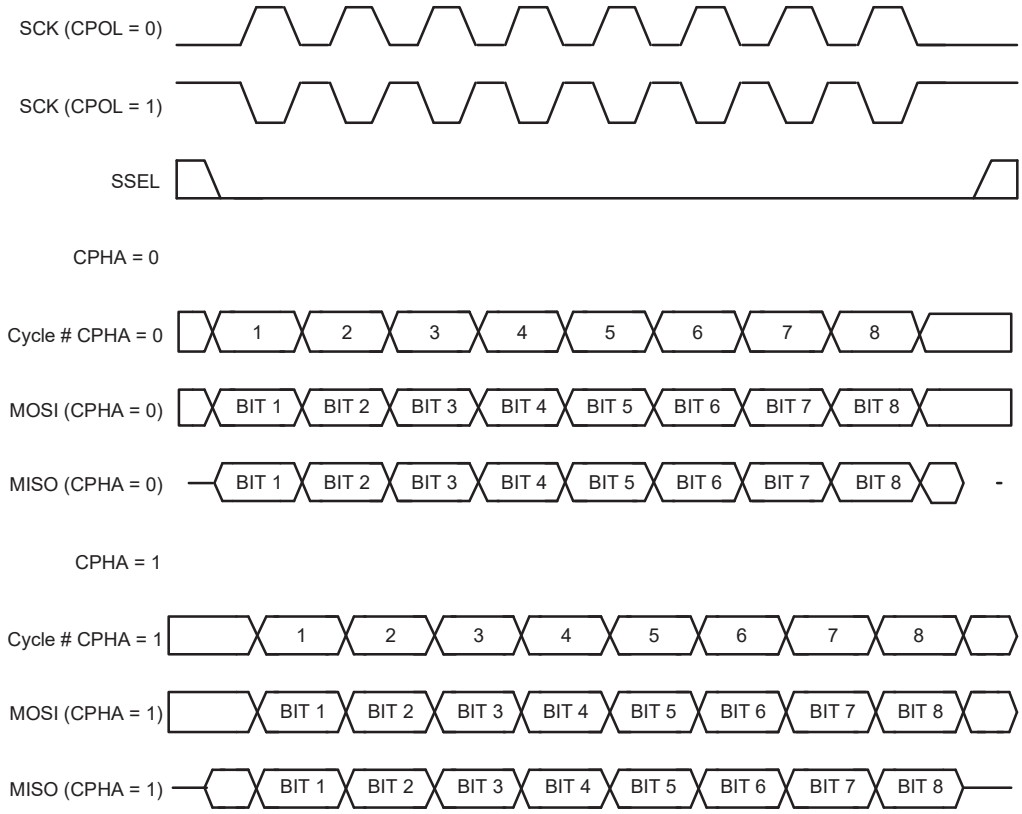

Fig 75. SPI data transfer format (CPHA = 0 and CPHA = 1)

The data and clock phase relationships are summarized in Table 360.

Table 360. SPI Data To Clock Phase Relationship

CPOL and CPHA settings	When the first data bit is driven	When all other data bits are driven	When data is sampled
CPOL = 0, CPHA = 0	Prior to first SCK rising edge	SCK falling edge	SCK rising edge
CPOL = 0, CPHA = 1	First SCK rising edge	SCK rising edge	SCK falling edge
CPOL = 1, CPHA = 0	Prior to first SCK falling edge	SCK rising edge	SCK falling edge
CPOL = 1, CPHA = 1	First SCK falling edge	SCK falling edge	SCK rising edge

The definition of when a transfer starts and stops is dependent on whether a device is a master or a slave, and the setting of the CPHA variable.

When a device is a master, the start of a transfer is indicated by the master having a byte of data that is ready to be transmitted. At this point, the master can activate the clock, and begin the transfer. The transfer ends when the last clock cycle of the transfer is complete.

When a device is a slave and CPHA is set to 0, the transfer starts when the SSEL signal goes active, and ends when SSEL goes inactive. When a device is a slave, and CPHA is set to 1, the transfer starts on the first clock edge when the slave is selected, and ends on the last clock edge where data is sampled.

17.6 SPI peripheral details

17.6.1 General information

There are five control and status registers for the SPI port. They are described in detail in Section 17.7 “Register description” on page 417.

The SPI Control Register (S0SPCR) contains a number of programmable bits used to control the function of the SPI block. The settings for this register must be set up prior to a given data transfer taking place.

The SPI Status Register (S0SPSR) contains read-only bits that are used to monitor the status of the SPI interface, including normal functions, and exception conditions. The primary purpose of this register is to detect completion of a data transfer. This is indicated by the SPI Interrupt Flag (SPIF) in the S0SPINT register. The remaining bits in the register are exception condition indicators. These exceptions will be described later in this section.

The SPI Data Register (S0SPDR) is used to provide the transmit and receive data bytes. An internal shift register in the SPI block logic is used for the actual transmission and reception of the serial data. Data is written to the SPI Data Register for the transmit case. There is no buffer between the data register and the internal shift register. A write to the data register goes directly into the internal shift register. Therefore, data should only be written to this register when a transmit is not currently in progress. Read data is buffered. When a transfer is complete, the receive data is transferred to a single byte data buffer, where it is later read. A read of the SPI Data Register returns the value of the read data buffer.

The SPI Clock Counter Register (S0SPCCR) controls the clock rate when the SPI block is in master mode. This needs to be set prior to a transfer taking place, when the SPI block is a master. This register has no function when the SPI block is a slave.

Prior to use, SPI configurations such as the master/slave settings, clock polarity, clock rate, etc. must be set up in the SPI Control Register and SPI Clock Counter Register.

The I/Os for this implementation of SPI are standard CMOS I/Os. The open drain SPI option is not implemented in this design. When a device is set up to be a slave, its I/Os are only active when it is selected by the SSEL signal being active.

17.6.2 Master operation

The following sequence can be followed to set up the SPI prior to its first use as a master. This is typically done during program initialization.

1. Set the SPI Clock Counter Register to the desired clock rate.
2. Set the SPI Control Register to the desired settings for master mode.

The following sequence describes how one should process a data transfer with the SPI block when it is set up to be the master. This process assumes that any prior data transfer has already completed.

1. Optionally, verify the SPI setup before starting the transfer.
2. Write the data to transmitted to the SPI Data Register. This write starts the SPI data transfer.

3. Wait for the SPIF bit in the SPI Status Register to be set to 1. The SPIF bit will be set after the last cycle of the SPI data transfer.
4. Read the SPI Status Register.
5. Read the received data from the SPI Data Register (optional).
6. Go to step 2 if more data is to be transmitted.

Note: A read or write of the SPI Data Register is required in order to clear the SPIF status bit. Therefore, if the optional read of the SPI Data Register does not take place, a write to this register is required in order to clear the SPIF status bit.

17.6.3 Slave operation

The following sequence can be followed to set up the SPI prior to its first use as a slave. This is typically done during program initialization.

1. Set the SPI Control Register to the desired settings for slave mode.

The following sequence describes how one should process a data transfer with the SPI block when it is set up to be a slave. This process assumes that any prior data transfer has already completed. It is required that the system clock driving the SPI logic be at least 8X faster than the SPI.

1. Optionally, verify the SPI setup before starting the transfer.
2. Write the data to transmitted to the SPI Data Register (optional). Note that this can only be done when a slave SPI transfer is not in progress.
3. Wait for the SPIF bit in the SPI Status Register to be set to 1. The SPIF bit will be set after the last sampling clock edge of the SPI data transfer.
4. Read the SPI Status Register.
5. Read the received data from the SPI Data Register (optional).
6. Go to step 2 if more data is to be transferred.

Note: A read or write of the SPI Data Register is required in order to clear the SPIF status bit. Therefore, at least one of the optional reads or writes of the SPI Data Register must take place, in order to clear the SPIF status bit.

17.6.4 Exception conditions

Read Overrun

A read overrun occurs when the SPI block internal read buffer contains data that has not been read by the processor, and a new transfer has completed. The read buffer containing valid data is indicated by the SPIF bit in the SPI Interrupt Register being active. When a transfer completes, the SPI block needs to move the received data to the read buffer. If the SPIF bit is active (the read buffer is full), the new receive data will be lost, and the read overrun (ROVR) bit in the SPI Status Register will be activated.

Write Collision

As stated previously, there is no write buffer between the SPI block bus interface, and the internal shift register. As a result, data must not be written to the SPI Data Register when a SPI data transfer is currently in progress. The time frame where data cannot be written to the SPI Data Register is from when the transfer starts, until after the SPI Status

Register has been read when the SPIF status is active. If the SPI Data Register is written in this time frame, the write data will be lost, and the write collision (WCOL) bit in the SPI Status Register will be activated.

Mode Fault

If the SSEL signal goes active when the SPI block is a master, this indicates another master has selected the device to be a slave. This condition is known as a mode fault. When a mode fault is detected, the mode fault (MODF) bit in the SPI Status Register will be activated, the SPI signal drivers will be de-activated, and the SPI mode will be changed to be a slave.

If the SSEL function is assigned to its related pin in the relevant Pin Function Select Register, the SSEL signal must always be inactive when the SPI controller is a master.

Slave Abort

A slave transfer is considered to be aborted if the SSEL signal goes inactive before the transfer is complete. In the event of a slave abort, the transmit and receive data for the transfer that was in progress are lost, and the slave abort (ABRT) bit in the SPI Status Register will be activated.

17.7 Register description

The SPI contains 5 registers as shown in Table 361. All registers are byte, half word and word accessible.

Table 361. SPI register map

Name	Description	Access	Reset Value[1]	Address
S0SPCR	SPI Control Register. This register controls the operation of the SPI.	R/W	0x00	0x4002 0000
S0SPSR	SPI Status Register. This register shows the status of the SPI.	RO	0x00	0x4002 0004
S0SPDR	SPI Data Register. This bi-directional register provides the transmit and receive data for the SPI. Transmit data is provided to the SPI0 by writing to this register. Data received by the SPI0 can be read from this register.	R/W	0x00	0x4002 0008
S0SPCCR	SPI Clock Counter Register. This register controls the frequency of a master's SCK0.	R/W	0x00	0x4002 000C
S0SPINT	SPI Interrupt Flag. This register contains the interrupt flag for the SPI interface.	R/W	0x00	0x4002 001C

[1] Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

17.7.1 SPI Control Register (S0SPCR - 0x4002 0000)

The S0SPCR register controls the operation of SPI0 as per the configuration bits setting shown in Table 362.

Table 362: SPI Control Register (S0SPCR - address 0x4002 0000) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	BitEnable	0	The SPI controller sends and receives 8 bits of data per transfer.	0
		1	The SPI controller sends and receives the number of bits selected by bits 11:8.	
3	CPHA		Clock phase control determines the relationship between the data and the clock on SPI transfers, and controls when a slave transfer is defined as starting and ending.	0
		0	Data is sampled on the first clock edge of SCK. A transfer starts and ends with activation and deactivation of the SSEL signal.	
		1	Data is sampled on the second clock edge of the SCK. A transfer starts with the first clock edge, and ends with the last sampling edge when the SSEL signal is active.	
4	CPOL		Clock polarity control.	0
		0	SCK is active high.	
		1	SCK is active low.	
5	MSTR		Master mode select.	0
		0	The SPI operates in Slave mode.	
		1	The SPI operates in Master mode.	
6	LSBF		LSB First controls which direction each byte is shifted when transferred.	0
		0	SPI data is transferred MSB (bit 7) first.	
		1	SPI data is transferred LSB (bit 0) first.	
7	SPIE		Serial peripheral interrupt enable.	0
		0	SPI interrupts are inhibited.	
		1	A hardware interrupt is generated each time the SPIF or MODF bits are activated.	
11:8	BITS		When bit 2 of this register is 1, this field controls the number of bits per transfer:	0000
		1000	8 bits per transfer	
		1001	9 bits per transfer	
		1010	10 bits per transfer	
		1011	11 bits per transfer	
		1100	12 bits per transfer	
		1101	13 bits per transfer	
		1110	14 bits per transfer	
		1111	15 bits per transfer	
		0000	16 bits per transfer	
31:12	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.7.2 SPI Status Register (S0SPSR - 0x4002 0004)

The S0SPSR register controls the operation of SPI0 as per the configuration bits setting shown in Table 363.

Table 363: SPI Status Register (S0SPSR - address 0x4002 0004) bit description

Bit	Symbol	Description	Reset Value
2:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
3	ABRT	Slave abort. When 1, this bit indicates that a slave abort has occurred. This bit is cleared by reading this register.	0
4	MODF	Mode fault. when 1, this bit indicates that a Mode fault error has occurred. This bit is cleared by reading this register, then writing the SPI0 control register.	0
5	ROVR	Read overrun. When 1, this bit indicates that a read overrun has occurred. This bit is cleared by reading this register.	0
6	WCOL	Write collision. When 1, this bit indicates that a write collision has occurred. This bit is cleared by reading this register, then accessing the SPI Data Register.	0
7	SPIF	SPI transfer complete flag. When 1, this bit indicates when a SPI data transfer is complete. When a master, this bit is set at the end of the last cycle of the transfer. When a slave, this bit is set on the last data sampling edge of the SCK. This bit is cleared by first reading this register, then accessing the SPI Data Register. **Note:** this is not the SPI interrupt flag. This flag is found in the SPINT register.	0
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.7.3 SPI Data Register (S0SPDR - 0x4002 0008)

This bi-directional data register provides the transmit and receive data for the SPI. Transmit data is provided to the SPI by writing to this register. Data received by the SPI can be read from this register. When used as a master, a write to this register will start an SPI data transfer. Writes to this register will be blocked when a data transfer starts, or when the SPIF status bit is set, and the SPI Status Register has not been read.

Table 364: SPI Data Register (S0SPDR - address 0x4002 0008) bit description

Bit	Symbol	Description	Reset Value
7:0	DataLow	SPI Bi-directional data port.	0x00
15:8	DataHigh	If bit 2 of the SPCR is 1 and bits 11:8 are other than 1000, some or all of these bits contain the additional transmit and receive bits. When less than 16 bits are selected, the more significant among these bits read as zeroes.	0x00
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.7.4 SPI Clock Counter Register (S0SPCCR - 0x4002 000C)

This register controls the frequency of a master's SCK. The register indicates the number of SPI peripheral clock cycles that make up an SPI clock.

In Master mode, this register must be an even number greater than or equal to 8. Violations of this can result in unpredictable behavior. The SPI0 SCK rate may be calculated as: PCLK_SPI / SPCCR0 value. The SPI peripheral clock is determined by the PCLKSEL0 register contents for PCLK_SPI as described in Section 4.7.3.

In Slave mode, the SPI clock rate provided by the master must not exceed 1/8 of the SPI peripheral clock selected in Section 4.7.3. The content of the S0SPCCR register is not relevant.

Table 365: SPI Clock Counter Register (S0SPCCR - address 0x4002 000C) bit description

Bit	Symbol	Description	Reset Value
7:0	Counter	SPI0 Clock counter setting.	0x00
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.7.5 SPI Test Control Register (SPTCR - 0x4002 0010)

Note that the bits in this register are intended for functional verification only. This register should not be used for normal operation.

Table 366: SPI Test Control Register (SPTCR - address 0x4002 0010) bit description

Bit	Symbol	Description	Reset Value
0	Test	SPI test mode. When 0, the SPI operates normally. When 1, SCK will always be on, independent of master mode select, and data availability setting.	NA
31:1	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.7.6 SPI Test Status Register (SPTSR - 0x4002 0014)

Note: The bits in this register are intended for functional verification only. This register should not be used for normal operation.

This register is a replication of the SPI Status Register. The difference between the registers is that a read of this register will not start the sequence of events required to clear these status bits. A write to this register will set an interrupt if the write data for the respective bit is a 1.

Table 367: SPI Test Status Register (SPTSR - address 0x4002 0014) bit description

Bit	Symbol	Description	Reset Value
2:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
3	ABRT	Slave abort.	0
4	MODF	Mode fault.	0
5	ROVR	Read overrun.	0

Table 367: SPI Test Status Register (SPTSR - address 0x4002 0014) bit description

Bit	Symbol	Description	Reset Value
6	WCOL	Write collision.	0
7	SPIF	SPI transfer complete flag.	0
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.7.7 SPI Interrupt Register (S0SPINT - 0x4002 001C)

This register contains the interrupt flag for the SPI0 interface.

Table 368: SPI Interrupt Register (S0SPINT - address 0x4002 001C) bit description

Bit	Symbol	Description	Reset Value
0	SPIF	SPI interrupt flag. Set by the SPI interface to generate an interrupt. Cleared by writing a 1 to this bit. **Note:** this bit will be set once when SPIE = 1 and at least one of SPIF and WCOL bits is 1. However, only when the SPI Interrupt bit is set and SPI0 Interrupt is enabled in the NVIC, SPI based interrupt can be processed by interrupt handling software.	0
7:1	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

17.8 Architecture

The block diagram of the SPI solution implemented in SPI0 interface is shown in the Figure 76.

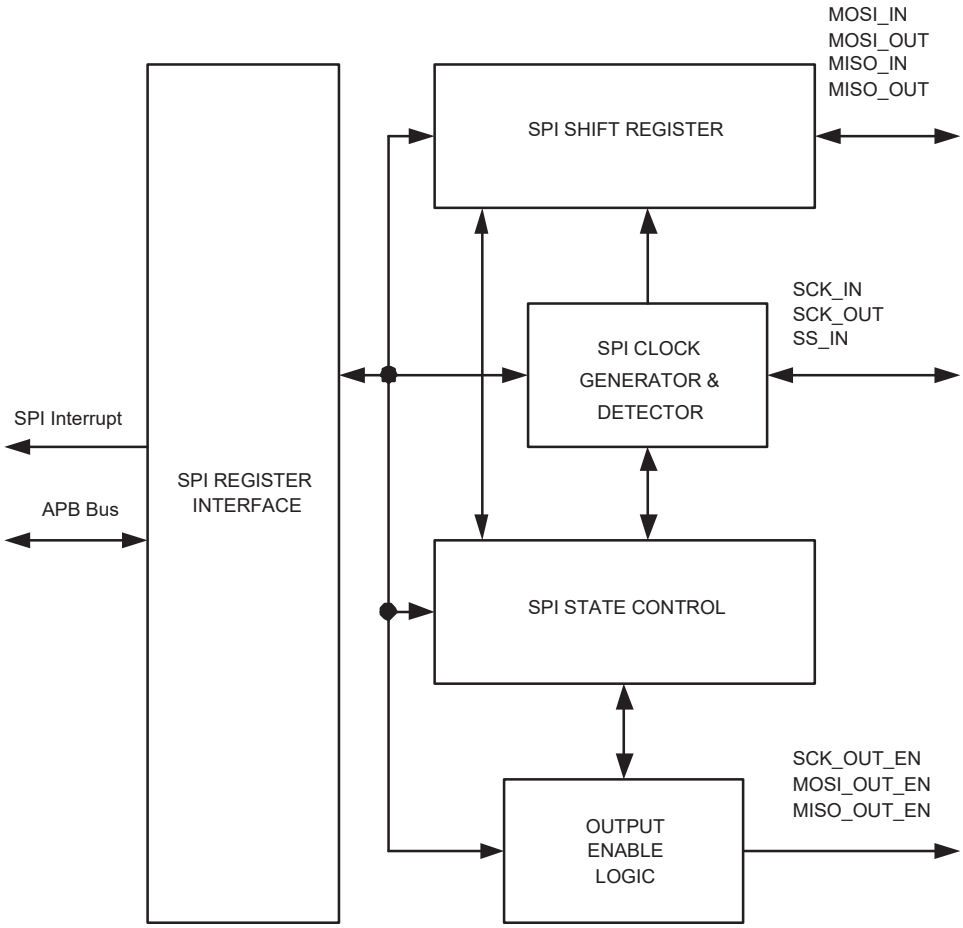

Fig 76. SPI block diagram